Programação Orientada a Objetos 2019/2020
Trabalho prático

Este enunciado é constituído por três partes:

Parte A - Regras do trabalho

O trabalho é constituído por um programa em C++, que deve implementar o tema proposto usando corretamente os princípios de orientação a objetos e os mecanismos e classes da biblioteca standard dados nas aulas. A interface com o utilizador é feita segundo o conceito de consola não-gráfica, e será disponibilizada uma biblioteca para auxiliar nesta tarefa. O tema proposto deixa diversos aspetos em aberto que são de relevância menor para o trabalho e devem ser resolvidos de acordo com a lógica e bom senso, desde que não removam trabalho ou complexidade ao tema.

O trabalho pode ser feito em grupos de 2 alunos, podendo ser alunos de laboratórios diferentes.

O trabalho é entregue em duas metas. As entregas serão feitas via moodle, e as indicações serão dadas na devida altura. Existem defesas que são obrigatórias e influenciam bastante a nota. As regras adicionais relativas ao trabalho prático encontram-se descritas na ficha da disciplina, cuja consulta é obrigatória antes da colocação de questões quanto a estas regras.

As questões omissas que sejam relevantes serão resolvidas e divulgadas pelos docentes no moodle.

Não considerando os conhecimentos prévios de unidades curriculares do 1º ano, este projeto requer os tópicos de programação C ++ que são abordados na unidade curricular e apenas esses.

Parte B - Descrição do tema

1. INTRODUÇÃO

Pretende-se um simulador de corridas de carros em C++. O simulador engloba alguns conceitos óbvios tais como ***Corrida***, ***Carro***, ***Piloto, Autódromo, Pista e Garagem*** (pode haver mais). Cada um destes conceitos é explicado abaixo. No entanto é muito importante que considere o seguinte:

- Apenas as principais entidades são descritas. Outras entidades, com maior ou menor relevância, poderão ser necessárias.
- Não existe necessariamente a relação uma entidade – uma classe. É muito provável que essa relação exista, mas pode acontecer que a representação de uma entidade/conceito seja distribuída por várias classes, ou apenas como atributo de outra.

- A descrição foca principalmente as propriedades e comportamentos de cada conceito/classe. Os detalhes específicos sobre métodos e dados, incluindo os que não são mencionados explicitamente, devem ser deduzidos e implementados.
- Os membros das classes devem ser deduzidos em função do comportamento desejado para cada classe, consoante o conceito que essa classe representa. Deve escolher tipos e estruturas de dados que façam sentido. Deve ter particular atenção em evitar estratégias que fariam sentido em C mas não em C++.
- Há mais do que uma estratégia e implementação possíveis.

2. CARROS

Um carro é um pequeno dispositivo que simula um veículo real com o seguinte comportamento:

- Possui marca e modelo. Nenhum carro pode ser criado sem marca. O modelo também deve ser definido no início, mas se nada for dito, terá o modelo "modelo base".
- Cada carro tem uma identificação (uma letra de ***a*** a ***z***) que é automaticamente atribuída na criação do carro. Se houver mais carros que letras, os excedentes ficam com a letra '?'. A atribuição da letra é automática (sem intervenção do resto do programa), sento o primeiro o 'a'.
- É elétrico e possui uma certa quantidade de energia (em mAh). Pode ser carregado, mas para isso deve estar parado. O processo de carregamento envolve uma manivela que fornece ***n*** mAh, sendo ***n*** maior que zero. O carro não permitirá ultrapassar a capacidade máxima da sua bateria. A capacidade máxima e a capacidade inicial são características especificadas quando um carro é fabricado. O valor ***n*** é indicado quando a operação é feita.
- Pode estar em movimento ou parado. Quando em movimento, avança a uma determinada velocidade (metros por segundo).
- Em movimento, um carro pode manter a sua velocidade, acelerar ou travar. O acelerador pressupõe dois usos: pressionar (o pedal do) acelerador e largar (o pedal do) acelerador. Enquanto o acelerador estiver pressionado, é exercido o seu efeito, que é de aumentar a velocidade do carro em 1m/s. O efeito é exercido apenas uma vez por segundo e mantém-se até o (pedal do) acelerador ser largado. Quer isto dizer que se quiser aumentar em 2m/s o acelerador terá que ser premido, mantido premido durante 2 segundos e depois largado, ou então premido e largado duas vezes. Pode-se premir e largar imediatamente, sendo exercido o seu efeito; no entanto, só se pode premir uma vez em cada segundo, e só se pode largar uma vez em cada segundo.
- O carro possui também um travão que tem um funcionamento exatamente nos mesmos moldes que o acelerador, sendo a única diferença a de diminuir a velocidade em 1m/s em vez de a aumentar. Tudo o resto é igual ao acelerador.
- Existe uma velocidade máxima, que é um dos parâmetros da construção de um carro. Essa velocidade máxima nunca será ultrapassada. De igual forma, a velocidade nunca será inferior a 0. Não é possível pressionar simultaneamente os pedais para acelerar e travar.

- Mover consome energia: 0,1 mAh por segundo por metro/s. Se não houver energia, o carro diminui a velocidade, como se estivesse a travar (continuamente, perdendo 1m/s por cada segundo). Se além disso o condutor travar, o seu efeito é somado ao efeito de perda de velocidade por falta de energia.
- Possui um sinal de emergência que pode ser ligado ou desligado, independentemente da velocidade.
- Um carro faz tudo o que foi descrito acima apenas se tiver um piloto (veja abaixo) – é o piloto que opera os vários mecanismos no carro em que está. Só pode estar um piloto dentro do carro. O piloto pode sair ou entrar no carro apenas quando este estiver parado.
- O carro deve ter uma maneira de saber que passou (mais) um segundo (tempo). Isso permitirá que ele simule o seu comportamento (por exemplo, gastar energia enquanto se move, acelerar, travar, etc.).
- Um carro pode ficar irremediavelmente danificado. Se isso acontecer, o carro fica imediatamente com velocidade 0 e deixará de responder a qualquer comando exceto os dois seguintes: tirar de um autódromo, continuando a existir (ver adiante) e remover de todo (deixar de existir). Se o carro tiver um dano irreparável, o piloto perde a vida (obviamente, se lá estiver um piloto no momento).
- Um carro danificado não aceita condutor.

3. PILOTOS

O piloto é um pequeno robô que simula o comportamento de um piloto real. Possui as seguintes características:

- Tem um nome, atribuído quando é fabricado (tal como as pessoas reais, quando nascem) e nunca muda.
- O nome de cada piloto é único. Este aspeto é verificado automaticamente pela classe que, se detetar repetição, modificará o nome dado automaticamente de forma a ser único.
- Pode estar a conduzir um carro, ou seja, estar dentro desse carro. Naturalmente, num dado instante só pode conduzir um carro. Pode operar os mecanismos do carro que está a conduzir. Isto implica que o piloto tenha a capacidade de decidir o que vai fazer com o carro, e também a capacidade de receber ordens (vindas do exterior) para fazer determinada ação. Por exemplo: o piloto pode decidir travar, mas também pode receber uma ordem para o fazer (imagine um agente da autoridade que o manda parar por excesso de velocidade – quer ver os documentos). Basicamente, tudo o que pode ser feito com o carro pode ser decidido pelo condutor ou por via de uma ordem vinda "de fora".
- Pode receber ordens para entrar num carro específico ou sair do carro em que se encontra.
- Pode estar a competir numa corrida. Ao conduzir, a cada segundo da corrida, o piloto exibe um determinado comportamento de acordo com sua personalidade que o poderá levar a fazer algo, tal como, operar os controlos do carro de uma certa maneira. A personalidade específica de um piloto depende do seu tipo, conforme descrito abaixo.

4. Crazy Driver

Este tipo de piloto exibe o seguinte comportamento ao competir:

- É desatento e só se lembra de começar a correr no Xº segundo da corrida (sendo X um número aleatório entre 1 e 5).
- A cada novo segundo, pergunta à pista qual é o seu lugar na corrida (ver abaixo). Se não estiver no primeiro ou no último lugar, acelera. Se estiver em primeiro, mantém a velocidade. Se estiver em último, fica de mau humor e trava.
- Se notar que perdeu lugares ("ficou mais para trás") desde o instante (segundo) anterior, pisa no acelerador de forma a aumentar a velocidade em 2 m/s (exceto se estiver no último lugar, conforme descrito acima). Se o carro ficar sem energia, entra em pânico e acende o sinal de emergência.
- Existe uma probabilidade de 5% deste condutor fazer algo que danifica irremediavelmente o automóvel. Quando isso acontece, o automóvel no lugar imediatamente atrás também fica irremediavelmente danificado (o condutor assusta-se e despista-se) e mais nenhum automóvel sofre com isso.

5. Piloto rápido

O piloto rápido exibe o seguinte comportamento:

- É extremamente ansioso e começa a correr no primeiro segundo da corrida.
- Acelera continuamente até ver que a energia restante do carro atingiu metade da sua capacidade máxima. Por precaução, a partir desse momento, ele só atuará o acelerador uma vez (para + 1m/s) a cada 3 segundos.
- Devido à sua ansiedade, a cada 10 segundos tem uma probabilidade 10% de sofrer um ataque de pânico, e, caso se verifique esta situação, ativa o botão de emergência.

6. Piloto surpresa

Defina e implemente outro tipo de piloto que se comporte de maneira diferente dos dois anteriores. Esse tipo de piloto deve ser implementado apenas pelo seu grupo (por outras palavras, não é nada boa ideia aparecerem pilotos surpresa iguais em grupos diferentes).

7. Direção geral de viação - DVG

É a entidade que tem o registo de todos os carros e pilotos e a responsável direta pela existência destes objetos. Será esta a entidade alvo de backup caso o utilizador assim o determine (ver na seção de comandos).

8. AUTÓDROMO: PISTA E GARAGEM

O Autódromo compreende a pista propriamente dita, onde os carros competem, e uma garagem, onde os carros estão temporariamente armazenados quando não estão a competir. Pista e garagem são conceitos autocontidos, mas não existem fora do contexto do autódromo em que estão inseridos. O autódromo tem um nome que lhe permite distinguir-se de outros autódromos. O nome de cada autódromo é único, sendo isto validado e corrigido automaticamente pela classe em questão. O Autódromo tem uma pista certificada para um máximo de N carros. O número N é um dos parâmetros da sua construção. A pista tem um determinado comprimento em metros, correspondendo à distância entre a linha de partida e a linha de chegada. Esta distância é uma característica inicial do autódromo e nunca poderá ser alterada, pois o autódromo já está construído e não é feito de borracha.

Os carros não pertencem ao autódromo e podem sair de um autódromo e irem para outro, para competirem noutras corridas. No entanto, se um autódromo for destruído (por exemplo, ocorreu uma inundação), todos os carros que estejam nele ficam irremediavelmente danificados e consideram-se automaticamente fora desse autódromo. Presume-se que é possível um carro não estar em nenhum autódromo a um dado instante. Estará então apenas sob a alçada da direção geral de viação, que é quem realmente controla a existência dos carros. Esta entidade também controla a existência dos pilotos.

A corrida decorre na pista, onde os carros são colocados para esse efeito. Quando uma corrida está a decorrer, a pista mantém o controle da posição e lugar de cada um dos carros que estão a competir. O carro estará numa determinada ***posição*** a partir do início da pista (o sítio onde está nesse momento), e estará num determinado ***lugar*** (na corrida), sendo o primeiro lugar o que vai mais à frente. Os veículos que se encontram na garagem não são contados para o máximo de N carros e não competem em corridas. A pista exibe o seguinte comportamento:

- A pista permite a adição de carros, desde que a corrida ainda não tenha começado. Os carros são colocados lado a lado. A pista é como a das corridas dos 100 metros olímpicos: existem N linhas, uma para cada carro.

- A pista permite a inserção de um piloto num carro. Obviamente, só terá efeito se o carro estiver vazio e parado, conforme as regras do carro.

- A pista permite que se comece a corrida. Isso só acontece se houver pelo menos dois carros concorrentes na pista e todos os veículos na pista tiverem condutor. Após o início da corrida, mais nenhum carro é aceite e nenhuma mudança de piloto é permitida.

- A pista permite fazer avançar o tempo em um ou mais segundos na corrida, se esta já tiver começado e não terminado. Cada vez que esse mecanismo é ativado, a pista indica a cada um dos carros que o tempo avançou. Note que no caso de se avançar mais do que um segundo de uma vez, os carros terão que ser informados explicitamente de cada um desses segundos para que o seu comportamento seja correto (exemplo, a aceleração e travagem).

- Se algum carro ligar o sinal de emergência, será removido da corrida e movido para a garagem. A garagem é um local que permite armazenar carros. Quando um carro é colocado na garagem, é indicado ao piloto para deixar o veículo, porque, para ele, a corrida acabou.

- Permite terminar a corrida. Os carros são removidos da pista e vão para a garagem, e os pilotos saem dos veículos.

- Possui um mecanismo para obter ("obter" ≠ "mostrar") a informação relacionada com uma corrida, indicando se ela já foi iniciada ou não, e, em caso afirmativo, detalhando a classificação atual. Um exemplo de classificação pode ser:

```
Informação sobre a corrida no autódromo Silverstone (5000 m):
1. B Ferrari / Vettel (rápido) - 5mAh, 290mAh - 2410m - 55m/s
2. D Mercedes / Hamilton (crazy) - 5mAh, 350mAh - 2300m - 50m/s
3.
...
```

9. CAMPEONATO

Existe ainda o conceito de campeonato:

- Um campeonato é composto por um conjunto de corridas, cada uma a decorrer num autódromo diferente, uma de cada vez. Portanto, Campeonato é um conceito por si só que agrega um conjunto de Autódromos nos quais decorre uma corrida em cada um deles por uma certa ordem.
- A associação pilotos/carros e os autódromos participantes são definidos antes do início do campeonato e não podem ser modificados a meio.
- Carros que fiquem irremediavelmente danificados numa corrida falham o resto do campeonato.
- No final de cada corrida é atribuída pontuação aos pilotos: 10 pontos para o primeiro classificado, 5 pontos para o segundo e 2 pontos para o terceiro. Estas pontuações só são atribuídas caso os pilotos tenham terminado a corrida.
- O campeonato deve manter uma classificação dos pilotos, permitindo mostrar o campeão e o vice-campeão quando as corridas terminarem.
- Existem vários aspetos e funcionalidades que devem ser analisados e definidos pelos alunos:
    - É necessário definir um conjunto de carros, pilotos e autódromos para um campeonato. Esta é a "população" para as várias corridas.
    - Devem existir mecanismos para efetuar a próxima corrida, ver pontuações, etc. (e outras funcionalidades óbvias).
    - Deve ser possível ao campeonato propagar comandos para os pistas, carros e pilotos.

Toda esta funcionalidade será acionada através de comandos do utilizador, descritos mais adiante.

Acerca da organização das várias entidades: já foi referido para a maior parte das entidades quem controla o quê / quem tem o quê. Para os casos omissos, pode assumir que o simulador é, em última análise, quem contém tudo.

10. VISUALIZAÇÃO

Deve implementar um mecanismo de visualização funcional, que permita:

- Ver a pontuação dos pilotos, ordenada de forma decrescente.
- Ver a corrida a decorrer, recorrendo a uma biblioteca auxiliar para controlo da consola, ou outra biblioteca qualquer à escolha dos alunos (exemplo: Qt, ncurses, outra). Não é preciso ter uma interface muito elaborada – apenas funcional e que permita distinguir os carros, ver onde estão de pelo seu posicionamento na pista, etc.
  - Visualmente, os carros são identificados por letras: esta deve ser minúscula sempre que um carro não está pilotado e maiúscula em caso contrário.
  - A pista é composta por várias "linhas", como as pistas das corridas dos 100 metros olímpicos. Cada carro tem o seu espaço (a sua linha no ecrã). Assuma que não irá haver demasiadas linhas para o ecrã.
  - A representação do carro na sua posição na pista pode ser feita de uma forma simples; através de uma regra de três simples, faz-se corresponder a dimensão da pista ao número de caracteres disponíveis no ecrã.
  - Os eventos que dependam de uma probabilidade, deverão ser registados num *log* no simulador. Este log poderá ser exibido no ecrã através de um comando (descrito mais adiante) .
- Deverá também ser possível visualizar os carros que se encontram na garagem, e neste caso, sem condutor.

A interação com o utilizador deverá ser centralizada na parte do programa onde faça mais sentido. Não se pretende ter carros, pilotos etc., todos a interagir com o utilizador. Em vez disso, os vários componentes do programa informação até à parte que realmente interage com o utilizador.

11. COMANDOS E INTERAÇÃO COM O UTILIZADOR

O utilizador interage com o simulador através de comandos escritos. Não está prevista a utilização de paradigmas gráficos e se o quiser fazer, pode, desde que não saia de C++, mas está por sua conta.

Cada comando é uma linha de texto formada pelo nome do comando e 0 ou mais parâmetros separados por espaços. Os parâmetros especificam qual o "objeto" ao qual se está a dar a ordem (carro, piloto, pista, autódromo, etc.) e restantes parâmetros, dependendo da ordem dada (exemplo: quantidade de segundos a avançar na simulação).

O simulador tem dois modos de funcionamento:
Modo 1 – Define o que existe: carros, autódromos, condutores, com todas as suas características. Também deve ser possível adicionar e retirar quaisquer destes elementos, assim como fazer entrar e sair pilotos em carros. Será possível fazer ainda, backup/recuperação para/da memória da direção geral de viação.
Modo 2 – Simulação de um campeonato: o simulador permite escolher os autódromos e a sua ordem no campeonato. Todos os carros que tenham piloto participam em todas corridas enquanto estiverem em condições de participar. Deve ser mantida uma pontuação para cada condutor.

Os caracteres **<** e **>** não serão escritos pelo utilizador, obviamente, e o que está entre esses < e > são valores ou nomes consoante o seu significado no contexto do comando.

Modo 1

No **modo 1**, o utilizador dispõe dos seguintes comandos:

- **carregaP <nomeFicheiro>** - Obtêm um conjunto de pilotos que deverão ser criados a partir dos dados presentes no ficheiro, cujas linhas têm o seguinte formato:

tipo nome

  O *tipo* poderá ser: crazy, rapido ou surpresa.

- **carregaC <nomeFicheiro>** - Obtêm um conjunto de carros que deverão ser criados a partir dos dados presentes no ficheiro, cujas linhas to seguinte formato:

capacidadeInicial capacidadeMaxima marca modelo

- **carregaA <nomeFicheiro>** - Obtêm um conjunto de autódromos a incluir neste campeonato que deverão ser criados a partir dos dados presentes no ficheiro, com o formato indicado abaixo. No caso dos autódromos pode assumir que o nome é apenas uma palavra.

N comprimento nome

- **cria <letraTipo> <dados do objeto>** - Acrescenta um objeto de um determinado tipo (c = carro, p = piloto, a = autódromo) onde *dados do objeto* são exatamente os dados que devem existir para se criar um objeto de uma dada classe (com o mesmo formato do respetivo ficheiro usado para criar vários objetos). Exemplo: *adiciona p rapido Jairton Senna* cria um piloto deste tipo com o nome indicado.

- **apaga <letraTipo> identificador** - Elimina um objeto de um determinado tipo (c = carro, p = piloto, a = autódromo) onde *identificador* é o nome que identifica os objetos, respetivamente: uma letra para identificar o carro (se o carro já estiver associado a um piloto, este fica livre para ser associado a outro carro); nome é o identificador para piloto e autódromo. Exemplo: *elimina p Jairton Senna* cria um piloto deste tipo com o nome indicado.

- **entranocarro <letraCarro> <nomePiloto>** - Faz o piloto nomePiloto entrar no carro nomeCarro. Se já estiver noutro carro, nada acontece, de acordo com o senso comum.

- **saidocarro <letraCarro>** - Faz o piloto que estiver no carro nomeCarro sair.

- **lista -** Mostra no ecrã a informação relativa aos carros, aos pilotos e aos autódromos, bem como quem está em que carro e mais o que for relevante saber acerca destas entidades.

- **savedgv <nome> -** Faz uma cópia do objeto que representa a Direção Geral de Viação para memória (atenção: não é para ficheiro), associando-o ao nome indicado. A DGV "original" continua a poder ser usada e manipulada pelos comandos.

- **loaddgv <nome>** - Recupera o objeto que representa a Direção Geral de Viação, previamente guardado em memória com o nome indicado. Todos os comandos afetarão esta cópia que foi recuperada e a que estava antes a ser considerada é perdida.

- **deldgv <nome>** - Apaga a cópia do objeto que representa a Direção Geral de Viação, em memória, associada ao nome indicado.

Modo 2

O modo 2 inicia-se com o comando ***campeonato*** no qual se define a ordem pela qual os autódromos participam no campeonato:

- **campeonato <A1> <A2> ... <An>** - A primeira corrida ocorre no autódromo *A1*, a segunda no *A2*, etc., onde ***An*** é o nome do autódromo ***n***. O número de parâmetros é equivalente ao número de autódromos que participam. As baterias de todos os carros são carregadas com a capacidade máxima das suas baterias.
- **listacarros** - Mostra a informação relativa aos carros.
- **carregabat <letraCarro>** <**Q**>- utiliza uma vez a manivela de carregamento de energia no carro identificado por *letraCarro* fornecendo ***Q*** mAh.
- **carregatudo** - carrega todo os carros ficando com as suas baterias no máximo.
- **corrida** - A corrida seguinte é iniciada. Só correm os carros que tenham pilotos e cujos carros estejam em condições de competir (por exemplo, não estão na garagem).
- **acidente <letraCarro>** - provoca um dano irreparável no carro identificado por *letraCarro*.
- **stop <nomePiloto>** - dá ordem para o carro conduzido pelo piloto *nomePiloto* parar (desacelerar até parar). Se estiver numa corrida sai dela. Se estiver num campeonato, sai da corrida mas mantém-se no campeonato e pode fazer as restantes corridas dessa campeonato.
- **destroi <letraCarro>** - remove o carro de existência (deixa mesmo de existir). Se o piloto estiver lá dentro, fica apeado.
- **passatempo <n>** - faz passar *n* segundos do tempo de simulação.
- **log** - mostra o log de mensagens que o simulador tem registado.

Parte C - Considerações, Metas e entregas

CONSIDERAÇÕES E REGRAS

É esperado o seguinte:

- O programa deve compilar sem nenhum *warning* ou erro.
- O utilizador (observação: "utilizador" ≠ "piloto") pode exibir um comportamento imprevisível e o programa deve ser razoavelmente robusto para esse aspeto (por exemplo, tolerar entradas de dados incorretas). Se o utilizador digitar valores incorretos (por exemplo, para a construção de objetos), uma exceção deverá ser gerada e capturada pelo programa.
- A simulação não deve causar erros de execução.
- Implemente todos os métodos necessários para tornar todas as suas classes robustas e autocontidas quanto ao seu uso em todas as situações habituais de C++.

ENTREGAS E DATAS IMPORTANTES

Meta 1 – Prazo: **24 de novembro de 2019**

Programa que implemente as seguintes funcionalidades:

- Interpretação de comandos
- Modo 1:
  - Execução de todos os comandos exceto savedgv, loaddgv e deldgv.
  - Cria -se apenas 1 autódromo.
  - Cria-se só um tipo de piloto, genérico (descrito na seção 3).
- Modo 2:
  - Comando "campeonato <A1>" (só há um autódromo).
  - Comando "passatempo <N>"
  - Nesta meta os carros movem-se sempre uma posição (metro / s) para a frente independentemente do piloto que o conduz. Não são carregados nem gastam energia.
- Visualização do movimento dos carros.
- Relatório – serão dadas indicações acerca do conteúdo do relatório.

Meta 2 – Prazo: **5 de janeiro de 2020**
Objetivos: o programa completo, com relatório.